

A Product Model based Approach to Interactive CAE Design Optimization

T. Fahrig, B. Nachtwey, S. Geller, J. Tölke, M. Krafczyk
Institut für Computeranwendungen im Bauingenieurwesen (CAB)
Technische Universität Braunschweig,
Pockelsstraße 3, 38106 Braunschweig, Germany
{fahrig/nachtwey/geller/toelke/kraft}@cab.bau.tu-bs.de

Summary

We present a software prototype for fluid flow problems in civil engineering, which combines essential features of Computational Steering approaches with efficient methods for model transfer and high performance computing.

The main components of the system are described:

- The modeler with a focus on the data management of the product model
- The pre-processing and the post-processing toolkit
- The simulation kernel based on the Lattice Boltzmann method
- The required hardware for real-time computing

1 Introduction

Functional Design of complex CAE objects requires a close co-operation of specialists in various disciplines. Especially in the early design phase, the effort to correct errors or inadequacies is relatively small. As the design goals of the different engineering disciplines often tend to be mutually exclusive, efficient communication between the engineering specialists is essential for an effective design process and mandatory for an economic optimization during the design and the economic lifetime of the construction. An automated computer-assisted system serving as a framework for this process would presuppose a consistent theoretical description of all partial models as part of an object and process oriented generalized theory about the interactions of all building components and processes. Such a theory is not yet available.

2 Application

The presented prototype describes a heuristic strategy: A group of engineers from different disciplines come together in an interactive and virtual design space representing an open-plan office and evaluate in close cooperation the mutual implications of their specific design variants online. Depending on the positioning of the office furniture and an HVAC (heat-conditioning, ventilation, air-conditioning) installation a human wellness index is interactively computed. The wellness factor based on the PMV- Index (Predicted Mean Vote; Ritschel, 1994), depending on the temperature, the density and the velocity of the airflow inside the office, as well as additional values defined by the user.

The engineers are provided the opportunity to locally or globally optimize the PMV-Index by moving objects and modifying parameters of the HVAC installation within the modeler. If the geometry or the boundary conditions change, the index will be recomputed automatically and the results are visualised in real time.

The optimization process is controlled by dedicated functions implemented in the modeling tool. If a modification exceeds a defined critical value, the system will inform the user or, in simpler cases, correct the modification.

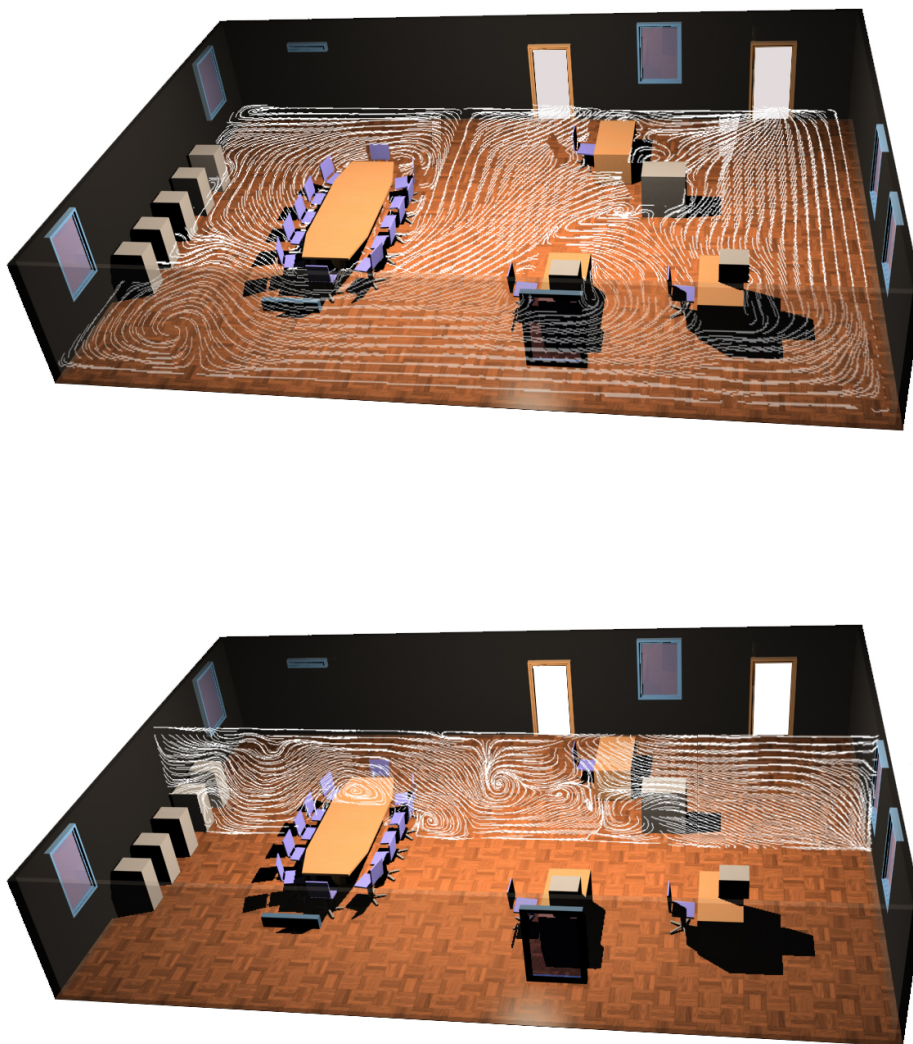


Fig. 1: Air-flow in an open-plant office

3 System layout

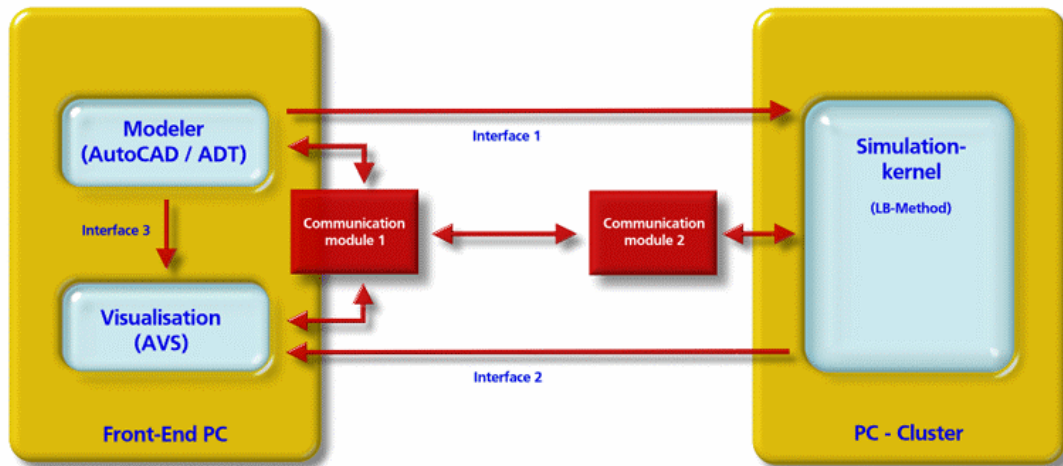


Fig. 2: System layout

3.1 Hardware

For modeling and visualisation we use a 64-Bit dual-CPU PC and a professional high performance graphics card. This PC is connected directly to a virtual reality environment where the numerical results are displayed.

The virtual reality environment is based on a stereoscopic visualisation system. Two beamers equipped with directed polarization filters project two images (one separate image from each of the two outputs of the graphics card) on a transparent screen. Polarization filters allow to separate the two images for the left and the right eye. For operating the Virtual Reality environment, a 3D motion tracker based on a magnetic system is used.

To obtain an acceptable response time between the modification of the geometry and the visualisation of the results (i.e. within a minute), high performance computing is required. Therefore, the simulation kernel runs on a PC Cluster equipped with Myrinet2000 network with a peak performance of 250 GFLOPS delivered by its 96 CPU's and approximately 150 GB RAM. An additional high-speed optical Gigabit network directly and exclusively connects the cluster and the Front-End PC.

The virtual design space and the visualisation can be modified by additional engineers using PCs connected to the Front-End-PC over the network or internet. This is possible by utilizing a multi-user- environment based on the freeware VNC. Additional software and hardware is used for audio- and video-communication.

3.2 Software Components

The prototype is based on the modules as illustrated in Figure 2. The virtual design space is generated in the modeler for which we use the Autodesk Architectural Desktop. The simulation kernel is a research CFD prototype implementation based on the Lattice-Boltzmann method and uses smoothed unstructured Cartesian octree meshes. For visualisation of the flow and other computed quantities the AVS/Express toolkit is used. All the modules are described in more detail in the following text.

As shown in Figure 2 additional communication modules and interfaces are required to allow direct network based data exchange between the modules. Their implementation is based on the TCP/IP socket communication protocol.

3.3 Modeler

The basis for the modeler is the CAD-system Autodesk Architectural Desktop (ADT), an extension of the AutoCAD software which can be regarded as the de-facto standard in civil engineering. This choice is motivated by the systems powerful modeler which comes with a powerful object-oriented programming interface allowing an efficient data exchange with the controlling process, the communication modules and allows direct access to a facet model of the object geometry. In addition, ADT can be utilized for the whole data management including import, export and conversion between the product model and other model representations.

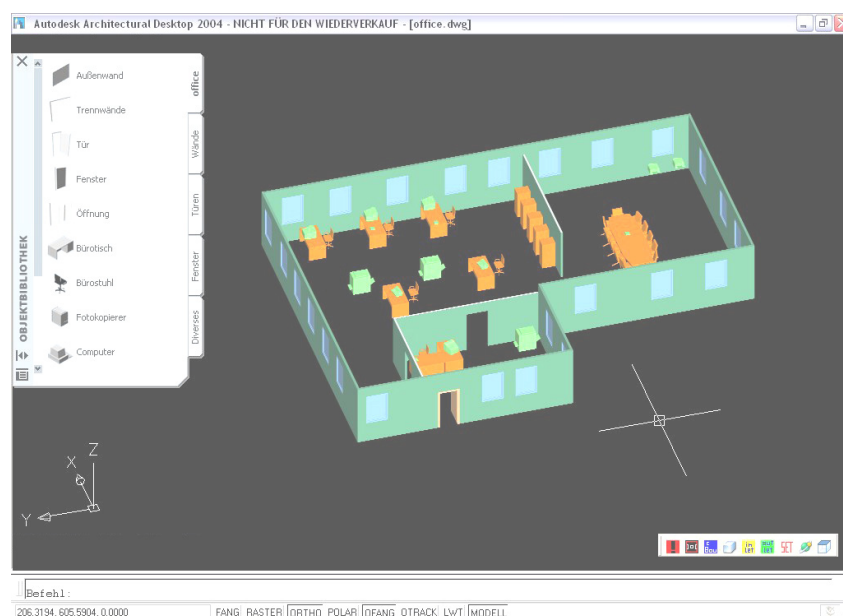


Fig. 3: Snapshot of a model developed using Autodesk Architectural Desktop (ADT)

Figure 3 shows the user interface of the virtual design space of the ADT. The options of modifying the geometry and the manipulation of the view are reduced to the essential functions for optimal usability. The user has the possibility to move, delete or add objects or change attributes for the simulation of special objects or object-types. For a fast workflow the user can drag a prepared object from an object library into the design space. New objects can be created externally and imported into the library during the work process.

Additional attributes are required for each object which provide information for the dedicated controlling functions. For example, a power socket may only be placed on a wall in a distance of 30 cm above the floor. Such built-in rules for the behaviour of objects are stored as additional object properties.

3.4 Data management

The challenge of the data management concept of our prototype is the keeping of all required data during the hole life cycle of the product model. Thus simulations can be run on every status of development or also for facility management purposes during the life time of the office building.

Generally, the following additional object data are required for the prototype:

1. Attributes for the fluid simulation, e.g. boundary condition attributes such as temperature, velocity, pressure or corresponding material properties such as thermal conductivity
2. Attributes for visualisation purposes, e.g. colour or transparency
3. Attributes for the control functions, i.e. the objects behaviour with respect to its position in the system.
4. The geometry of each object defined as a consistent volume model in order to allow an automatic mapping of a surface representation to an octree mesh for the numerical simulation kernel.

For an efficient data management the modeler has to be fully compatible with the product model standard including the import and export of data with additional object attributes required for the simulation process.

The latest version of ADT does not support product models out-of-the-box. With a special plug-in (IFC-Utility 2x for ADT 2004) the ADT is able to work with product model compliant data based on the Industry Foundation Classes (IFC) of the IAI. The IFC are the most open product modeling technology for the building construction today, but object attributes required for numerical simulations such as temperature, velocity or surface properties are not yet included in the IFC product model. To extend this product model with additional data, lists of variables and attributes can be attached to every component. The IfcPropertySets defined in the IfcXML format offer such a functionality.

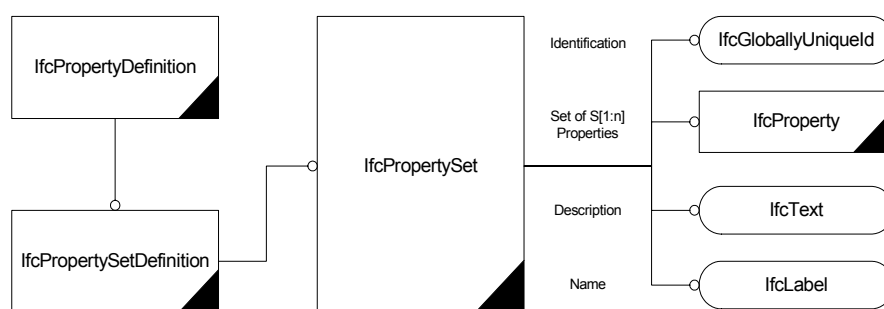


Fig. 4: Diagram of the IfcPropertySet

The property set is a container class that holds properties within a property tree. An identifier, a name and additional text describe each property set assigned to an object. Each set can have an unlimited list of properties.

The Autodesk AEC Object Modeling Framework (OMF), the C++ Runtime Extension program of Autodesk ADT, can access the data of the IFC product model. All additional attributes needed for the prototype and attached to an OMF-object are imported or exported as IfcPropertySets. This also includes the attributes for the behaviour of an object with respect to its location in the system. This is based on the use of OMF anchor elements that allow moving

an object to specific positions or its behaviour towards other elements. For example, an anchor can be defined in its geometry as a point, a line, a surface or a volume and acts like a spatial limiter for the corresponding object it is attached to.

The user has direct access to these attributes within ADT by OMF menus.

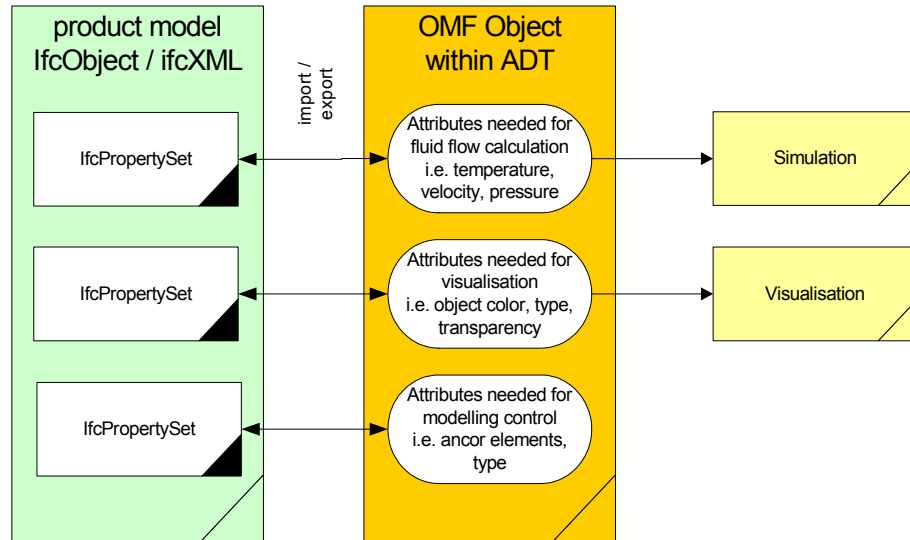


Fig. 5: product model data mangement

The ADT programming interface allows direct access to the object geometry and can transform it into a Brep-model (3D volume model, see Autodesk ADN). This model is based on facets and is required for the automatic mesh generation of the fluid solver. The robustness and efficiency of the automatic mesh generation process is ensured by additional functions integrated within ADT to control the consistency of this model conversion without gaps and overlaps.

4 Pre-Processing and CFD-Kernel

The simulation process itself does not receive any product model specific data. Only geometric objects and their additional physical attributes for fluid simulation are sent from the modeler to the pre-processor.

The pre-processing is an octree based grid generator which converts the facet model data to a grid required by the CFD-kernel. This automatic 3D-grid generation substantially optimizes the interactive workflow of the prototype and allows short design cycles.

The attributes of the facet model objects are adapted to the corresponding nodes of the CFD-grid and will trigger a consistent behaviour in the CFD-kernel.

4.1 Grid Generation and Octree

Hierarchical octree grids are widely used and powerful approaches to model fluid flows in complex 3D domains. Here we use a generalized octree component approach with base components for modeling physical and geometric problems based on (Brüggemann, 2004).

By a straight-forward extension with desired additional components this general octree can be extended to store all necessary nodal quantities for the CFD-kernel.

4.2 CFD-Kernel and communication interfaces

The CFD-kernel consists of three major components which obtain information from the modeler, distribute and collect the data and do the computation. They are implemented as a set of threads communicating with each other using the message passing interface (MPI). The first component is written in ANSI-C, because the operating system of the Modeler is a 32 bit Windows whereas the CFD-kernel is running on 64 bit Linux. To the best of our knowledge, no standard communication-toolkit allows the automatic mapping between these environments, so we use basic socket communication which is only available in ANSI-C or C++.

One part of the automatic grid generator is also included in this thread because it transforms the real coordinates of each data point into grid coordinates. For optimum performance the other components are written in Fortran90. The second component contains the second/remaining part(s) of the grid generator transforming object properties into computational properties of the grid nodes. It also distributes these data to the corresponding computing nodes. Thirdly this thread collects the data and sends it as a single dataset to the visualisation module using the Visualisation Interface Toolkit (Visit) developed at the Research Centre Juelich. The main effort is spent on the third component that is responsible for the computation of the fluid dynamics including the energy equation.

The fluid dynamics is computed by a Lattice-Boltzmann approach using the so-called D3Q19 model which includes interaction with 18 neighbouring nodes, the energy computation is based on a Finite-Difference method using the same 19-point star. For both parts the required parallel communication is accelerated by asynchronous communication which means that non-blocking communication routines are used. First the boundaries of each subdomain are treated and while waiting for the incoming data the inner nodes of each subdomain are computed.

4.2.1 Lattice-Boltzmann method

It is well known for many decades (Chapman 1990), that the Navier-Stokes equations

$$(1) \quad \frac{\partial \vec{u}}{\partial t} + (\vec{u} \nabla) \otimes \vec{u} = -\frac{1}{\rho} \nabla p + \nu \Delta \vec{u}$$

and the continuity equation $\nabla \vec{u} = 0$ (here both written for an incompressible fluid) can be derived from the Boltzmann equation

$$(2) \quad \frac{\partial f}{\partial t} + \vec{v} \nabla f = \Omega$$

where f is the probability to find a particle with a microscopic velocity \vec{v} at (\vec{x}, t) given a suitable collision operator Ω describing the influence of particle collisions. The connection between the microscopic quantity f and the macroscopic fields is defined via moments of the form $\Phi_i = \int \vec{v}^i f d\vec{v}$ where $\Phi_0 = \rho$ and $\Phi_1 = \rho \vec{u}$ and for weakly compressible flows by an equation of state $p(\rho, T) = 0$. Thus in principle it is possible to obtain solutions for the Navier-Stokes equations by solving the Boltzmann equation. As this latter equation is much harder to solve in general, there is nothing to gain from such an approach. Fortunately, it turns out that the Boltzmann equation can be drastically simplified and still can act as a hyperbolic superset of the Navier-Stokes equations. For this purpose one can discretize the microscopic velocity space of the particles (Grad, 1949), i.e. the continuous variable \vec{v} is replaced by a set of discrete, constant velocities $\{\vec{e}_j\}$, $j = 0, \dots, b$ which typically span a space filling unit cell in two or three dimensions.

Thus we obtain a set of coupled equations of the form

$$(3) \quad \frac{\partial f_i}{\partial t} + \vec{e}_i \nabla f_i = \Omega_i(\{f_j\})$$

where $i, j = 0, \dots, b$. This set of equations is referred to as the discrete Boltzmann equation due to its discreteness of the microscopic velocity space. Another substantial simplification is to replace the collision operator by the so-called BGK approximation, which merely includes the tendency of the system to develop towards an equilibrium state.

$$(4) \quad \frac{\partial f_i}{\partial t} + \vec{e}_i \nabla f_i = -\frac{1}{\tau}(f_i - f_i^{eq})$$

where τ is a microscopic relaxation time. For an ideal gas the famous Maxwell distribution can be chosen as the equilibrium function. Macroscopic quantities can now be obtained from summations in analogy to the moments obtained via integration as above:

$$(5) \quad \rho = \sum_i f_i \text{ and } \rho \vec{u} = \sum_i f_i \vec{e}_i.$$

A straight forward approach to solve the discrete Boltzmann equations is to use an explicit first order upwind Finite Difference scheme on a uniform grid with grid spacing Δx . One obtains the following set of algebraic relations:

$$(6) \quad f_i(\vec{x} + \vec{e}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t) = -\frac{\Delta t}{\tau}(f_i(\vec{x}, t) - f_i^{eq}(\vec{x}, t))$$

This equation is frequently referred to as the Lattice-BGK-equation or Lattice-Boltzmann equation. Alternative discretization techniques for eq. (4) have been developed optimized for specific problems.

5 Post-Processing and Visualisation

AVS/EXPRESS (AVS, 2004) is used for the visualisation of the systems physical behaviour as computed by the CFD kernel. This program includes a powerful programming interface and many modules are available to expand its built-in library functions for visualisation and communication with other programs.

Even for large datasets AVS allows an efficient visualisation of scalar and vector fields. Isosurfaces, isovolumes and streamlines are utilized to display the airflow and the distribution of the PMV-Index inside the office.

A special AVS module allows to import the CAD geometry (via DXF files) to be directly imported into AVS. The extracted dataset based on a vector format including additional visualisation attributes such as colors and transparency.

The data of the simulation kernel and the modeler are combined within AVS and visualised.

6 Conclusion

A prototype of a computational steering environment as the one described above can substantially reduce the time for many complex optimization processes which require human expertise during the design process. The use of thermal simulations in general can reduce the life cycle costs of a building up to 5% (Pierre Barles, 2000). This calculation includes the cost

of the HVAC system and its consumption costs. Especially in the design phase of complex buildings with a high budget, functional optimization by CFD analysis will likely gain more importance. This development will be accelerated by further progress with respect to computer hardware and efficient numerical techniques the combination of which will offer reasonable capabilities to predict the behaviour of complex engineering systems in the early design phase.

7 References

- Autodesk Architectural Desktop / AutoCAD, <http://www.autodesk.com>
- Autodesk Architectural Desktop - OMF Developer's Guide 3.3, Autodesk Developer Network, <http://adn.autodesk.com>
- Autodesk ObjectARX Developer's Guide, Autodesk Developer Network, <http://adn.autodesk.com>
- AVS/EXPRESS, <http://www.avs.com>
- Barles (2000): *Etude de l'usage de la Simulation Dynamique dans la Conception des Bâtiments en France*, PBC
- Brüggemann, Holz (2004): *Quadtree based hydroinformatics simulation systems: A component-oriented finite volume toolkit for shallow water equations*, 6th International Conference on Hydroinformatics (submitted)
- Chapman, Cowling (1990): *The Mathematical Theory of Non-Uniform Gases*. Cambridge, University Press
- Grad (1949): *On the Kinetic Theory of Rarified Gases*, Communications on Pure and Applied Mathematics, pp. 331-407
- IAC – International AVS Centre, <http://www.iavsc.org/>
- IFC-Utility 2x for ADT 2004 - G.E.M. Team Solutions, <http://www.team-solutions.de/>
- IFC / ifcXML, IAI – International Alliance for Interoperability, <http://www.iai.org.uk>
- Kühner, Krafczyk, Tölke, (2002): *Towards interactive comfort optimization of Indoor Flows using Virtual Reality based analysis of Large-Eddy simulation results*, in proceedings of ICCCBE-IX, The 9th International Conference on Computing in Civil and Building Engineering, Taipei, Taiwan, pp. 843-848
- MPI-Forum, <http://www.mpi-forum.org>
- Neuberg, Hoffmann (2001): *Simulation von Energieströmen in Gebäuden auf der Grundlage eines IFC basierten Produktmodells*, proceedings Forum Bauinformatik
- Ritschel (1994), *Raumklimatechnik*, Band 1, Springer Verlag, Berlin, pp.125-153
- Tölke, Fahrig, Nachtwey, Krafczyk (2003): *Computational Steering in Civil Engineering*, IKM - Internationales Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen Weimar
- Visit at FZ Jülich, <http://www.fz-juelich.de/zam/visit>
- VNC Software by RealVNC, <http://www.realvnc.com>